УДК 519.161

М. А. Негодаев

ПРОБЛЕМА ВЫПОЛНИМОСТИ ЛИНЕЙНОЙ КАРТЫ ДЛЯ ИГР «САПЕР» И «ЦВЕТНОЙ САПЕР»

1. Введение

В работе рассматривается игра «Сапер» (Minesweeper) и ее модификация «Цветной сапер», связанная с разбиением всех закрытых клеток на классы и частичной расстановкой чисел на открытых клетках. Доказано, что проблема выполнимости карты ограниченной ширины ℓ для игр «Сапер» и «Цветной сапер» разрешима за полиномиальное время при $\ell \in \mathbb{N}$ и $\ell = 1$ соответственно. Показано, что проблема выполнимости карты ограниченной ширины $\ell > 1$ игры «Цветной сапер» является NP-полной.

Наличие известных, содержательных, просто формулируемых задач среди полных проблем в том или ином классе вычислительной сложности является несомненным стимулом для изучения этого класса. Кроме того, наличие таких проблем является принципиально важным и с точки зрения самого процесса изучения, поскольку эти проблемы предоставляют удобный инструментарий для расширения информации о классе посредством нахождения новых полных проблем интерпретацией, а также для построения быстрых алгоритмов. Рассмотрение комбинаторных игр приводит к внушительному ряду весьма интересных проблем в теории алгоритмов и теории вычислительной сложности. Удивительно, что многие кажущиеся сравнительно простыми головоломки и игры оказываются весьма сложными с вычислительной точки зрения. Эти игры доставляют интересные, просто и ясно формулируемые примеры полных проблем для различных классов вычислительной сложности (см. обзоры результатов в книге [1] и работе [2]). Так, многие игры для двух игроков являются PSPACE-полными (см. для примера [3, 4]). Такие известные игры, как шахматы и го, являются примерами EXPTIME-полных проблем (см. [5, 6, 7]). Классическая компьютерная игра Minesweeper – пример NP-полной проблемы (см. [8]). В рамках данной статьи мы будем использовать одно из ее русских названий – «Сапер».

2. Игра «Сапер» и некоторые ее вариации

Игра «Сапер» стала популярной во многом благодаря тому, что реализовывалась во всех версиях стандартных пакетов операционной системы Microsoft Windows (начиная с версии 3.0 и до самой последней включительно). Уже после появления первой реализации в операционной системе Microsoft Windows 3.0 были сделаны многочисленные попытки переноса игры на другие платформы и ее реализации с модифицированными правилами. Поскольку интерес к подобной деятельности не ослабевает и по сей день, к настоящему времени образовалось уже весьма обширное семейство «саперподобных» игр.

В игре «Сапер», реализованной в операционной системе Microsoft Windows, предлагается три вида карт: «новичок», «любитель» и «профессионал», имеющих различные размеры и количество мин. Эти параметры также могут быть заданы самим пользователем, причем максимальные размеры карты – 24×30, а максимальное количество мин – 667. Цель игры состоит в том, чтобы как можно быстрее найти все мины на минном поле, не вскрыв ни одну из них. Если открываемая ячейка содержит мину, вы проиграли.

◇ Игрок имеет возможность открыть любую ячейку, щелкнув по ней левой кнопкой мыши. Если открываемая ячейка содержит мину, вы проиграли.

◇ Если мины нет, в ячейке появится цифра, которая указывает, сколько мин находится в восьми смежных с ней ячейках.

◇ Игровое поле состоит из счетчика мин, секундомера и минного поля.

Помимо огромного количества поклонников, игра «Сапер» имеет также большое число реализаций, отличающихся не только интерфейсом, но и правилами. Например, клетки карты могут иметь различную форму: от правильных треугольников и шестиугольников до произвольной. Соответственно, и форма карты может быть различной, как и ее размеры. Теоретически размеры карты могут быть сколь угодно большими, а на практике они ограничиваются лишь техническими возможностями конкретного компьютера. При этом главным ограничением является количество свободной памяти. Существуют такие версии игры «Сапер», в которых размеры карты ограничены лишь объемом свободной оперативной памяти компьютера. Особый интерес представляет трехмерная модификация игры «Сапер». Эта игра реализуется как на трех полях, представляющих собой фронтальную, горизонтальную и вертикальную проекции трехмерного куба, так и на различных трехмерных фигурах.

Имеются также и другие модификации игры, не связанные с формой и размерами карты. К таким модификациям игры «Сапер» относится, например, игра Combat Engineer. В этой игре задача игрока состоит не в том, чтобы найти все мины, а в том, чтобы в наикратчайшее время открыть некоторые помеченные клетки карты. Чтобы до них добраться, необходимо расчистить путь от закрытых клеток, под которыми могут находиться мины. Кроме того, в игре Combat Engineer имеется возможность «подрываться» несколько раз, которая, впрочем, нередко встречается и в реализациях обыкновенной игры «Сапер». Существуют и такие версии игры, в которых все мины имеют различные «мощности», что требует принципиально иной стратегии разминирования.

Еще одной модификацией игры «Сапер» является игра Fox Hunt. Задача этой игры состоит в том, чтобы при минимальном количестве ходов в наикротчайшее время определить местонахождение всех лис, прячущихся на частично открытой карте. Выбирая ту или иную клетку, игрок либо находит лису, либо узнает суммарное количество лис, расположенных на одной горизонтали, вертикали и диагоналях относительно выбранной клетки.

Перечисленные вариации представляют далеко не полный список «саперподобных» игр. Этот список постоянно пополняется новыми играми и вариациями старых.

3. Проблема выполнимости карты

«Сапер» является игрой с неполной информацией. Отсюда, в частности, следует, что раскрываемость произвольной карты, вообще говоря, не гарантирована. Но в некоторых реализациях такие ситуации исключены, что обеспечивает существование беспроигрышной стратегии.

Одной из ключевых проблем при создании карт для игры является обеспечение их выполнимости. Решение этой же проблемы лежит в основе стратегии разминирования. Наша работа посвящена рассмотрению проблемы выполнимости карты для игры «Сапер» и одной из ее разновидностей, называемой нами «Цветной сапер». В этой игре все закрытые клетки имеют, вообще говоря, свой цвет и во всех клетках одного цвета мины одновременно либо есть, либо нет. Кроме того, функция, ставящая в соответствие открытым клеткам карты количество мин в соседних закрытых клетках, является частичной. Очевидно, что игра «Сапер» – это частный случай игры «Цветной сапер» и отличается следующими дополнительными условиями: цвета любых двух закрытых клеток различны и функция, ставящая в соответствие открытым клеткам карты количество мин в соседних закрытых клетках, всюду определена.

Игра «Цветной сапер» – типичный представитель классического семейства задач о принятии решения на основе сопоставления локальной и глобальной информации. Примером может послужить задача геологоразведки.

Для решения этой задачи необходимо сопоставить данные, поступающие со спутника, с данными от геологических партий. Данные со спутника – это глобальная информация, которая представляется на карте раскрашенными закрытыми клетками. Каждый цвет означает определенный тип земной поверхности. Информация от геологических партий – это числа на открытых клетках, которые говорят о количестве полезных ископаемых в окрестности данной клетки. Наличие мины в закрытой клетке определенного цвета определят наличие полезного ископаемого на данном участке карты. При этом считается известной вероятность наличия полезного ископаемого для каждого типа почвы. В результате такой интерпретации для решения задачи геологоразведки применяется вероятностный подход при разминировании карты игры «Цветной сапер». Еще одной важной интерпретацией игры «Цветной сапер» является задача ориентирования на местности. В ней аналогичным образом глобальные данные, получаемые с помощью GPS, сопоставляются с данными непосредственного наблюдения местности. Кроме того, задача игры «Цветной сапер» может использоваться при построении систем искусственного интеллекта для сопоставления значения функции энергии и значения ошибки на конкретной обучающей выборке. Можно привести множество подобных примеров, таких как поиск вирусов в Сети, дезактивация зараженной местности, но это не является нашей основной целью.

Возвращаясь к исследованию задач игр «Сапер» и «Цветной сапер», введем следующее определение. Будем говорить, что карта является выполнимой, если на ней можно разместить мины в закрытых клетках так, что их количество вокруг каждой открытой клетки, которой поставлено в соответствие число, будет равно числу, стоящему в самой клетке.

Если размеры карты зафиксированы, то проблема выполнимости очевидным образом разрешима за константное время. Если ограничений на размеры карты нет, то, как показано в [8], проблема выполнимости карты игры «Сапер», а значит, и игры «Цветной сапер» является NP-трудной. Для того чтобы проверить, выполняет ли карту данная расстановка мин, необходимо сравнить количество мин, расставленных вокруг каждой открытой клетки, с числом, стоящим в этой клетке, и проверить выполнение условия: либо все закрытые клетки одного цвета содержат мины, либо ни одна из них не содержит мины. Это сравнение требует $O(n^2)$ времени, следовательно, проблема выполнимости карты игры «Цветной сапер» является NP-полной. В связи с этим представляется естественным рассмотрение случая, когда один из размеров карты зафиксирован, а второй может быть произвольным. Такую карту будем называть линейной. В дальнейшем мы будем предполагать, что у линейной карты размера $\ell \times k$ ширина ℓ фиксирована, а длина k является переменной величиной. Проблемы выполнимости в случае линейной карты для игр «Сапер» и «Цветной сапер» формулируются следующим образом.

Проблема выполнимости линейной карты в игре «Сапер» ($MS(\ell)$)

Дано: *Частично открытая линейная карта разминирования размерами $\ell \times k$, где ℓ – постоянная. Все клетки карты разбиты на два класса: открытые и закрытые. На всех открытых клетках карты расставлены числа $0, \ldots, 8$, соответствующие количеству мин, расположенных в клетках, смежных с данной открытой клеткой. На закрытых клетках могут находиться мины.*

Задача: *Является ли карта выполнимой?*

Проблема выполнимости линейной карты в игре «Цветной сапер» ($CMS(\ell)$)

Дано: *Частично открытая линейная карта разминирования размерами $\ell \times k$, где ℓ – постоянная. Все клетки карты разбиты на два класса: открытые и закрытые. На некоторых открытых клетках карты расставлены числа $0, \ldots, 8$, соответствующие количеству мин, расположенных в клетках, смежных с данной открытой клеткой. На закрытых клетках могут находиться мины. Все закрытые клетки покрашены. При этом если две закрытые клетки имеют одинаковый цвет, то на них одновременно или есть мины, или их нет.*

Задача: *Является ли карта выполнимой?*

Для доказательства теорем в данной работе мы будем использовать классическую комбинаторную задачу SAT (выполнимость). Напомним ее формулировку.

Проблема выполнимости логической формулы (SAT)

Дано: *Логическая формула в КНФ.*

Задача: *Существует ли для данной формулы выполняющий набор значений переменных?*

Известно, что задача SAT является NP-полной. Часто эта задача рассматривается в «ограниченном» варианте N-SAT, в котором все конъюнкты формулы содержат не более N переменных. Известно, что задача 2-SAT решается за полиномиальное время, в то время как 3-SAT NP-полна.

4. Основные результаты

Теорема 1. *Задача $CMS(\ell)$ является NP-полной тогда и только тогда, когда $\ell>1$.*

Доказательство. Рассмотрим карту единичной ширины и покажем, что проверка ее выполнимости производится за полиномиальное время. Для этого проинтерпретируем задачу CMS(1) в задаче 2-SAT. Пусть $X = \{x_1, \ldots, x_t\}$ – множество обозначений цветов закрытых клеток. Заметим, что каждая открытая клетка карты из задачи CMS(1) имеет не более двух закрытых смежных с нею клеток. Следовательно, если какая-либо открытая клетка содержит число, большее двух, карта невыполнима. Исходя из этого, в дальнейшем будем предполагать, что в открытых клетках расставлены числа из множества $\{0, 1, 2\}$.

Приступим к построению формулы из задачи 2-SAT. Пробегая один раз по всем открытым клеткам карты, дополняем формулу новыми конъюнктами по следующим правилам:

$\diamond$ если текущая открытая клетка содержит число 0, то добавляем в формулу два конъюнкта $\overline{x}_i$ и $\overline{x}_j$ в том случае, если клетку окружают две закрытые клетки, а x_i и x_j – их цвета, или один конъюнкт $\overline{x}_i$ в случае, когда клетку окружает одна закрытая клетка цвета x_i;

$\diamond$ если текущая открытая клетка содержит число 1, добавляем в формулу конъюнкт $x_i \vee x_j$ в том случае, если клетку окружают две закрытые клетки, а x_i и x_j – их цвета, или конъюнкт x_i в случае, когда клетку окружает одна закрытая клетка цвета x_i;

$\diamond$ если текущая открытая клетка содержит число 2, то добавляем в формулу два конъюнкта x_i и x_j, где x_i и x_j – цвета двух смежных закрытых клеток; если же нашу открытую клетку окружает только одна закрытая клетка, то карта невыполнима.

Построенная в результате формула выполнима тогда и только тогда, когда существует такой набор значений ее переменных, который выполняет каждый конъюнкт, т. е. когда верна расстановка мин вокруг каждой открытой клетки. Иными словами, число мин вокруг каждой открытой клетки совпадает с числом, стоящим в самой клетке. Кроме того, поскольку каждая переменная принимает ровно одно значение из данного набора, все клетки одинакового цвета должны одновременно либо содержать мины, либо ни одна из них не должна содержать мин. В итоге получаем, что выполнимость построенной нами формулы эквивалентна выполнимости данной карты.

Заметим, что каждый конъюнкт формулы строится за константное время, а всего конъюнктов – $O(n)$, где n – размер формулы. Следовательно, построение формулы по карте производится за линейное время. Поскольку каждый конъюнкт нашей формулы состоит не более чем из двух элементов, формула удовлетворяет условиям задачи 2-SAT. Таким образом, мы за линейное время свели задачу CMS(1) к задаче 2-SAT. Это значит, что наша проблема CMS(1) лежит в классе P.

Теперь рассмотрим карту ширины два. Заметим, что если к ней добавить $k \in \mathbb{N}$ горизонталей, состоящих из пустых клеток, то мы получим карту ширины $k+2$. Напомним, что мы имеем дело с частичной картой, т е. функция, ставящая в соответствие открытой клетке число мин в смежных клетках, является частичной. Поэтому проведенное дополнение карты пустыми клетками не изменит свойства выполнимости карты. Учитывая все вышесказанное, докажем лишь NP-полноту задачи CMS(2). Для этого покажем сначала, что задача NP-трудна.

Доказательство трудности задачи CMS(2) будем вести интерпретацией задачи 3-SAT. Прежде чем осуществить эту интерпретацию, введем необходимые обозначения. Частично открытую карту можно отождествлять с матрицей, элементы которой принадлежат множеству $G = \{0, 1, \ldots, 8, ?\} \cup X$, где 0 обозначает открытую клетку, не смежную ни с одной закрытой клеткой, i, где $i \in \{1, 2, \ldots, 8\}$, обозначает открытую клетку, такую, что на закрытых клетках, смежных с данной, ровно i мин, «?» обозначает открытую клетку, такую, что информация о количестве мин вокруг нее не известна. Множество X суть множество обозначений для закрытых клеток в зависимости от их цветов, клетки одного цвета имеют одинаковые обозначения.

Для сведения 3-SAT к CMS(2) будем интерпретировать формулу

$$\bigwedge_{i=1}^{n} (a_i \vee b_i \vee c_i) \tag{1}$$

из задачи 3-SAT в виде матрицы M, представляющей частично открытую раскрашенную карту высоты два. Заметим, что формула (1) состоит из конъюнктов, имеющих следующий вид, с точностью до обозначений переменных и их перестановки:

$$a \vee b \vee c;$$
$$a \vee b \vee \overline{c};$$
$$a \vee \overline{b} \vee \overline{c};$$
$$\overline{a} \vee \overline{b} \vee \overline{c}.$$

Для начала построим матрицу для каждого из описанных конъюнктов и покажем, что произвольный набор значений переменных конъюнкта удовлетворяет его тогда и только тогда, когда соответствующая этому набору расстановка мин выполняет карту, представленную соответствующей конъюнкту матрицей. Для простоты последующих рассуждений все матрицы имеют размер 2×10.

Договоримся, что клетка карты, соответствующая одному из элементов матрицы d_{ij}, $\overline{a}_i$, $\overline{b}_i$, $\overline{c}_i$, имеет свой уникальный цвет. Тогда матрицы, интерпретирующие указанные выше дизъюнкции, выглядят следующим образом:

$$a_i \vee b_i \vee c_i : \quad \begin{pmatrix} a_i & b_i & c_i & ? & ? & ? & ? & ? & ? & ? \\ d_{i1} & 3 & d_{i2} & ? & ? & ? & ? & ? & ? & ? \end{pmatrix},$$

$$a_i \vee b_i \vee \overline{c_i} : \quad \begin{pmatrix} a_i & b_i & \overline{c}_i & ? & \overline{c}_i & 1 & ? & ? & ? & ? \\ d_{i1} & 3 & d_{i2} & ? & c_i & 1 & ? & ? & ? & ? \end{pmatrix},$$

$$a_i \vee \overline{b}_i \vee \overline{c}_i : \quad \begin{pmatrix} a_i & \overline{b}_i & \overline{c}_i & ? & \overline{b}_i & 1 & ? & \overline{c}_i & 1 & ? \\ d_{i1} & 3 & d_{i2} & ? & b_i & 1 & ? & c_i & 1 & ? \end{pmatrix},$$

$$\overline{a}_i \vee \overline{b}_i \vee \overline{c}_i : \quad \begin{pmatrix} a_i & b_i & c_i & ? & ? & ? & ? & ? & ? & ? \\ d_{i1} & 2 & d_{i2} & ? & ? & ? & ? & ? & ? & ? \end{pmatrix}.$$

Условимся, что логической переменной придается значение *true* в предположении, что мина в соответствующих клетках данного цвета есть, и *false* в противном случае. Покажем теперь, что выполнимость каждой из построенных карт на конкретной расстановке эквивалентна удовлетворимости конъюнкта на соответствующем расстановке наборе истиностных значений.

Рассмотрим для примера вторую дизъюнкцию. Она удовлетворена тогда и только тогда, когда либо одна из переменных a_i, b_i принимает значение *true*, т. е. хотя бы в одной из соответствующих клеток стоит мина, либо $c_i = false$, тогда клетка c_i не содержит мины. Вторая матрица как раз и реализует описанный случай: если мина находится в какой-то одной из клеток a_i или b_i, то, поскольку количество всех мин равно трем, остальные две мины содержатся в клетках d_{i1} и d_{i2}; если среди клеток a_i, b_i содержатся в точности две мины, то третья может стоять в d_{i1} или d_{i2}; если же, наконец, клетка c_i не содержит мины, то, следовательно, содержит мину клетка $\overline{c}_i$, тогда оставшиеся две мины скрываются за клетками a_i, b_i, d_{i1}, d_{i2}.

Таким образом, действительно выполнимость карты, представленной второй матрицей на конкретной расстановке, равносильна выполнимости второго конъюнкта на соответствующем расстановке наборе значений. Проверка оставшихся трех матриц производится аналогично, перебором всевозможных вариантов, связанных с расстановками мин в клетках a_i, b_i, c_i, таких, что рассматриваемая дизъюнкция удовлетворена.

Приступим теперь к построению карты, интерпретирующей формулу (1). Каждой переменной формулы (1) поставим в соответствие свой уникальный цвет. В качестве обозначения для клетки, имеющей этот цвет, будем использовать символ соответствующей переменной. Кроме того, нам потребуются еще цвета для раскраски вспомогательных закрытых клеток d_{ij}, $\overline{a}_i$, $\overline{b}_i$, $\overline{c}_i$. Таким образом, множество X обозначений раскрашенных клеток есть объединение двух множеств: множества C обозначений переменных формулы (1) и множества D обозначений вспомогательных закрытых клеток, каждая вспомогательная клетка имеет свой уникальный цвет, множества D и C не пересекаются.

В итоге построено отображение F, ставящее в соответствие конъюнкту C_i формулы (1) матрицу $M_i = (S_{i_1}, \ldots, S_{i_{10}})$, где S_{i_j} – j-й столбец матрицы M_i. Каждая из матриц M_i моделирует некоторый участок карты, в котором закрытые клетки играют роль логических переменных, причем одной переменной соответствуют клетки с одинаковым цветом. Формула (1) принимает истинное значение тогда и только тогда, когда каждый из конъюнктов удовлетворен. Положим

$$F(\varphi) \equiv M = (S_{1_1}, \ldots, S_{1_{10}}, \ldots, S_{n_1}, \ldots, S_{n_{10}}).$$

Получаем отображение $F\colon \Phi \to M[G]$, где Φ – множество всех формул вида (1), а $M[G]$ – множество всех матриц над G, ставящее в соответствие формуле φ матрицу M, представляющую карту минного поля. Очевидно, что преобразование F осуществляется за линейное время относительно числа конъюнктов n. Таким образом, проблема выполнимости формулы (1) сводится к проблеме существования расстановки мин, удовлетворяющей карте, т. е. задача 3-SAT проинтерпретирована в задаче CMS(2). А это означает, что проблема CMS(2), а вместе с ней и CMS(ℓ), $\ell \in \{2, 3, 4, \ldots\}$, – NP-трудные. Но поскольку любая индивидуальная задача проблемы CMS(ℓ), $\ell \in \{2, 3, 4, \ldots\}$, может быть проверена за линейное время путем последовательного сопоставления числа смежных мин каждой открытой клетки с числом, стоящим в самой клетке, то CMS(ℓ), $\ell \in \{2, 3, 4, \ldots\}$, является NP-полной проблемой.

Теорема 2. *При всех натуральных значениях ℓ задача* MS(ℓ) *разрешима за линейное время относительно размера карты.*

Доказательство. Дана линейная карта размера $\ell \times k$, ℓ – константа. Для доказательства теоремы опишем алгоритм проверки выполнимости данной карты и покажем, что этот алгоритм работает за $O(k)$. Для простоты изложения будем отождествлять элементы матрицы с клетками карты.

Рассмотрим матрицу $M_{\ell\times k}$, соответствующую данной карте. Пусть i – наименьшее из чисел, таких, что i-й столбец матрицы M содержит хотя бы одну открытую клетку. Обозначим j-й столбец матрицы M как S_j и рассмотрим матрицу $M_i = (S_{i-1} S_i S_{i+1})$. Если $i = 1$, то каждый элемент первого столбца матрицы M_i положим равным пустой открытой клетке «?», информация о количестве мин вокруг которой неизвестна. Заменим все открытые клетки в первом и третьем столбце матрицы M_i на пустые открытые клетки «?». Заметим, что первый столбец матрицы M_i не содержит открытых клеток, однако для матриц M_s, где $s > i$, это может не выполняться. Пусть A_i – множество всех расстановок для полученной матрицы. Их количество не превышает некоторой константы, поскольку M_i имеет размеры $\ell \times 3$, где ℓ – константа. Далее рассматриваем следующий столбец с наименьшим номером, содержащий хотя бы одну открытую клетку. Пусть это будет столбец S_j. Для него существует матрица M_j, для которой аналогичным образом строится множество A_j возможных расстановок. Затем если обе построенные матрицы имели общие столбцы (S_i и S_{i+1} или только S_{i+1}), то во множестве A_j оставляем только те расстановки, для которых существует расстановка во множестве A_i, совпадающая с данной на общих столбцах. Выбирая следующий столбец S_ℓ, если $\ell \leq j + 2$, мы будем искать общие расстановки во множествах A_j и A_ℓ.

Выполнение алгоритма продолжается до тех пор, пока мы не достигнем последнего столбца S_k, т. е. пока $i \leq k$. Если столбец S_k содержит хотя бы одну открытую клетку, то третий столбец матрицы M_k полагаем состоящим из пустых открытых клеток. Если на каком-то шаге A_i окажется пустым, значит, карта невыполнима. В противном случае карта является выполнимой.

Поскольку на каждом из $O(k)$ шагов алгоритма за постоянное время выполняется построение множества A_i, получаем, что описанный алгоритм работает за линейное время относительно размера карты.

Следствие. *Проблема* $\mathrm{MS}(\ell)$ *для карты размера* $\ell \times k$ *разрешима за время* $O(k \cdot 2^{3\ell})$.

Доказательство. Действительно, при выполнении алгоритма для карты размера $\ell \times k$ не более чем k раз выполняется построение множества A_i. При этом мощность множества A_i не превышает количества всех возможных расстановок мин на 3ℓ закрытых клетках. Таким образом, построение множества A_i производится за время $O(2^{3\ell})$. Следовательно, сложность описанного алгоритма $O(k \cdot 2^{3\ell})$.

В доказательстве теоремы 2 мы применяли алгоритм построения логической формулы по карте единичной ширины. Этот алгоритм имеет важность не только в рамках данной теоремы. Он позволяет использовать богатый арсенал быстрых алгоритмов, имеющийся для задачи 3-SAT, при решении задачи «Цветной сапер». Приведем неформальное описание этого алгоритма для карты произвольной ширины.

Алгоритм построения логической формулы по карте игры «Цветной сапер»

Для каждой клетки, содержащей число от 0 до 8, строим логическую формулу следующим образом. Выполняем перебор всех возможных расстановок мин на клетках, окружающих цифру. Количество клеток не превышает 8. Для каждой такой расстановки формируется своя конъюнкция, состоящая максимум из восьми переменных или их отрицаний, где каждая переменная представляет клетку некоторого цвета. Отрицание ставится или не ставится в зависимости от того, содержит клетка мину или нет. Полученные конъюнкции и являются дизъюнктами некоторой дизъюнктивной нормальной формы (ДНФ). Таким образом, для каждой клетки с числом от 1 до 8 имеем ДНФ такую, что количество расстановок мин вокруг клетки равно количеству наборов истиностных значений для ДНФ. Построенные ДНФ преобразуем в конъюнктивные нормальные формы. Тогда искомой формулой будет конъюнкция всех построенных формул.

Теорема 3. *Алгоритм построения логической формулы по карте игры «Цветной сапер» работает в логарифмическом пространстве относительно размера карты* n.

Доказательство. Для того чтобы двигаться по карте в поисках клетки, содержащей число, нам необходим счетчик клеток, который потребует $O(\log n^2)$ памяти. Кроме того, нужно вести нумерацию переменных в соответствии с нумерацией цветов, а это еще $O(\log n^2)$. Этого достаточно, чтобы каждому найденному числу на карте ставить в соответствие формулу, переменные которой соотнесены с цветами клеток на карте. Поскольку размеры построенных ДНФ ограничены некоторой константой, преобразование ДНФ к КНФ происходит за константное время. В итоге получаем, что построенный алгоритм требует $O(\log n)$ памяти.

Литература

1. Касьянов В. Н., Евстигнеев В. А. Графы в программировании: построение, визуализация и применение. СПб.: БХВ-Петербург, 2003.

2. Demaine E. Playing games with algorithms: Algorithmic combinatorial game theory // Lect. Notes in Comp. Sci. 2001. Vol. 2136. P. 18–32.

3. Bremner D., O'Rourke J., Shermer T. Motion planning amidst movable square blocks is PSPACE complete // Draft. 1994. June.

4. Culberson J. Sokoban is PSPACE-complete // Proc. Int. Conf. Fun with Algorithms. Waterloo, 1998. P. 65–76.

5. Fraenkel A. S., Lichtenstein D. Computing a perfect strategy for n x n chess requires time exponential in n // J. Combin. Theory. Ser. A. 1981. № 31. P. 199–214.

6. Robson J. M. The complexity of Go // Proc. IFIP 9th World Computer Congress. North-Holland, 1983. P. 413–417.

7. Robson J. M. N by N checkers is EXPTIME complete // SIAM J. Comput. 1984. Vol. 13, № 2. P. 252–267.

8. Kaye R. Minesweeper is NP-complete // Math. Intelligencer. 2000. Vol. 22, № 2. P. 9–15.